MOVIE GENRE PREDICTION USING PLOT SUMMARIES

Avantika Balaji, Charitha Uppalapati, Arthi Sri, Sanjay Balaji P, Sherwin Akshay J G
3rd year - V semester, B.Tech Computer Science and Engineering,
Amrita School of Computing, Coimbatore
Tamil Nadu, India

1. ABSTRACT

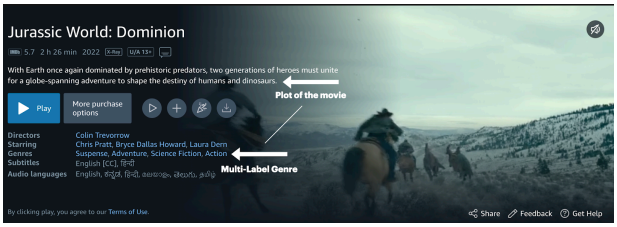
With the rise of digitalization comes the need to automatically label the ever-increasing digital content. Nonetheless, most video content is still labelled manually by users using textual descriptors or tags. However, in the last decade, the rise of machine learning methods and their application in various problem areas has spanned across various research domains, including video labelling. Our goal in this project is to investigate current state-of-the-art methods and then create our own movie genre classification solutions using various machine learning models. Because movie plots are widely available and can be assigned to genres, they are an excellent starting point for research into automatic text classification. Using the plot of the movie, we performed single-label and multi-label movie trailer genre classification.

2. INTRODUCTION

A typical binary classification problem is classifying a textual document into one of two pre-defined classes. A data instance in a multi-class classification problem is associated with only one of the many single class labels. In our project, we will look into a scenario in which each document can be assigned to more than one class, also known as multi-label classification.
Supervised text classification is a mature tool that has seen significant success in a variety of applications. When it comes to movies, most previous research has focused on predicting movie reviews or revenue, with little research done on predicting movie genres. Movie genres are still tagged manually, with users sending suggestions to The Internet Movie Database's email address (IMDB). Because a plot summary conveys a lot of information about a movie, we have explored different machine learning methods to classify movie genres using synopsis in this project.

3. PROBLEM STATEMENT

The goal of our project is to create a model that can predict the genre of a movie based solely on the plot details. In this work, we provide various model architectures that can be used to predict the genre of a movie across various films present in our dataset based on the synopsis.



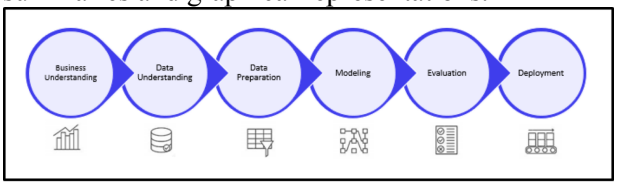
4. DATASET

Our dataset acquired from Kaggle is all about Movies/TV Shows that are available on Netflix. It consists of 9 columns: movie title, cast, brief description of the plot, duration, rating on IMDB, voted by people, year, genre, certificate. There is a total of 7912 unique movie titles. All data is taken from IMDB website by web scraping. We have ensured that the data collected, while small, is accurate and obtained from reputable movie review websites. We believe that a small but strong and correct dataset is preferable to a large and noisy dataset.

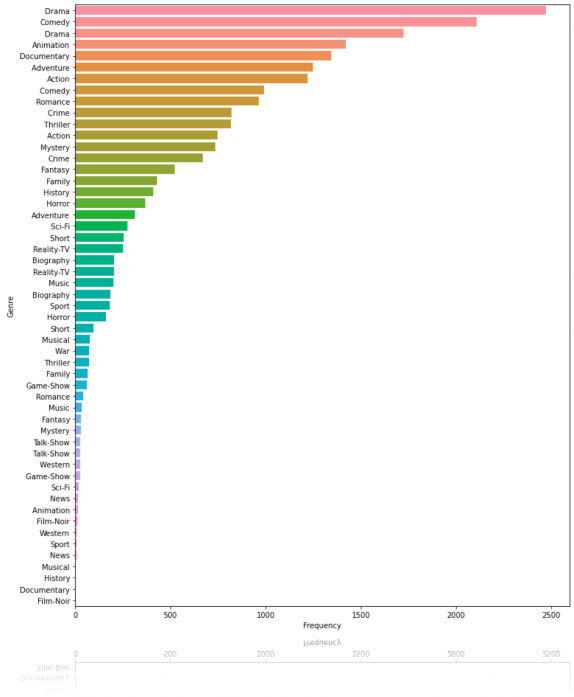
5. DATA PREPARATION

5.1 EXPLORATORY DATA ANALYSIS (EDA)

EDA is a method of analysing data that employs visual techniques. It is used to discover trends, patterns, or to validate assumptions using statistical summaries and graphical representations.



Here is the frequency distribution of each genre. A total of 53 genres are present in our dataset.



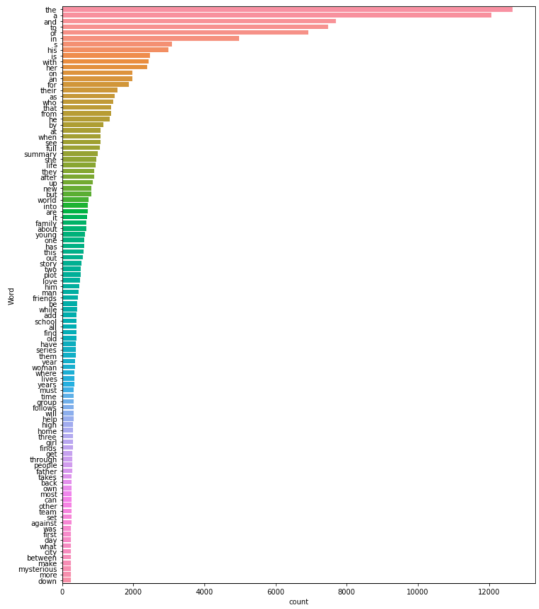
5.2 DATA PRE PROCESSING USING NLP

1. TEXT CLEAN UP

We cannot fit a raw text directly into a machine learning or deep learning model. Hence, we must clean the text first, which means splitting it into words along with handling punctuation and case. By manual tokenisation, we can load the data, remove forward slashes, which removes everything except letters. This can be achieved using the re.sub() function which belongs to the Regular Expression Module (re)

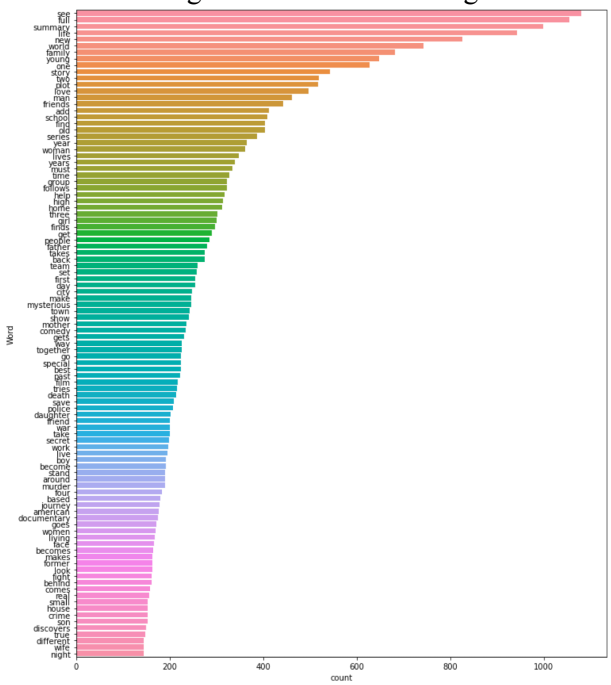
2. FREQUENTLY USED WORDS

Most frequently used words in the description of movies



3. REMOVE STOP WORDS

Stop words are English words that add little meaning to a sentence. They can be safely ignored without affecting the sentence's meaning.



4. ENCODING TEXT TO BINARY

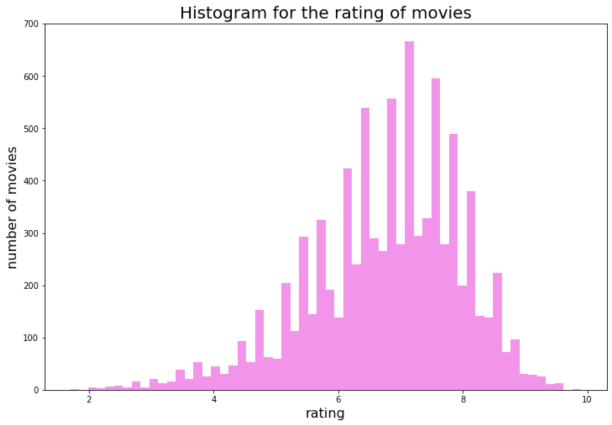
To run our machine learning algorithm on our categorical data, the data must first be converted to numerical data. One-hot encoding is one of the techniques used to perform this conversion. It is a vector representation of words in vocabulary. Categorical variables are represented as binary vectors in one-hot encoding. These categorical values are first converted to integers.
For this, we use sklearn's MultiLabelBinarizer()

5. EXTRACTING FEATURES USING TF-IDF

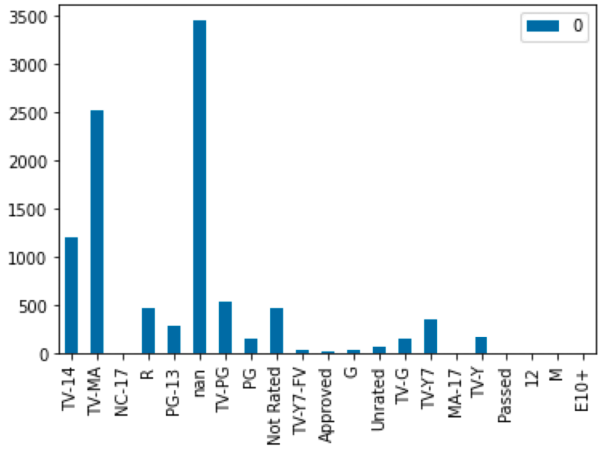
TF-IDF which stands for Term Frequency – Inverse Document Frequency. It is one of the most important techniques used for information retrieval to represent how important a specific word or phrase is to a given document.

5.3 DATA VISUALISATION

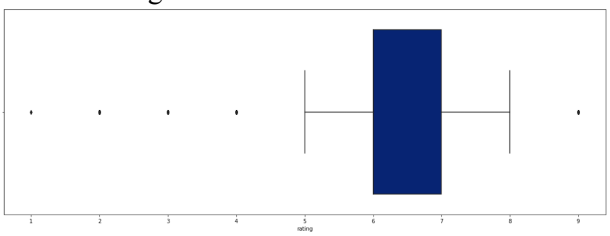
1. Histogram plot for ratings of movies



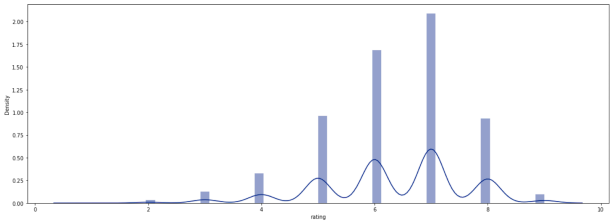
2. Bar plot of number of movies with each certificate type



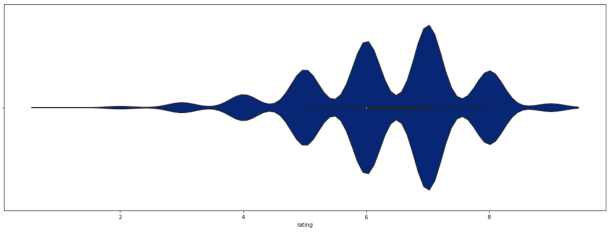
3. Box Plot, Distplot and Violin plot for movie ratings



BOX PLOT



DISTPLOT



VIOLIN PLOT

6. MODEL BUILDING

Next we split our data into train and validation sets for training and evaluating our model's performance. We have done a 80-20 split, with 80% data samples in the train set and the rest in validation set.

STANDARD METHODS FOR SOLVING MULTI-LABEL CLASSIFICATION:

Problem Transformation Method

In this method, we will try to transform our multi-label problem into single-label problem(s).
This method can be carried out in three different ways as:

1. Binary Relevance
2. Classifier Chains
3. Label Powerset

1. **Binary Relevance**- This problem is broken into 53 different single class classification problems as shown in the figure below. (53 genres)

X	Y_1
$x^{(1)}$	0
$x^{(2)}$	1
$x^{(3)}$	0
$x^{(4)}$	1
$x^{(5)}$	0

X	Y_2
$x^{(1)}$	1
$x^{(2)}$	0
$x^{(3)}$	1
$x^{(4)}$	0
$x^{(5)}$	0

X	Y_3
$x^{(1)}$	1
$x^{(2)}$	0
$x^{(3)}$	0
$x^{(4)}$	0
$x^{(5)}$	0

X	Y_4
$x^{(1)}$	0
$x^{(2)}$	0
$x^{(3)}$	0
$x^{(4)}$	1
$x^{(5)}$	1

2. **Classifier Chains** - The first classifier is trained just on the input data and then each next classifier is trained on the input space and all the previous classifiers in the chain.

X	y1	y2	y3	y4
x1	0	1	1	0
x2	1	0	0	0
x3	0	1	0	0

X	y1
x1	0
x2	1
x3	0

Classifier 1

X	y1	y2
x1	0	1
x2	1	0
x3	0	1

Classifier 2

X	y1	y2	y3
x1	0	1	1
x2	1	0	0
x3	0	1	0

Classifier 3

X	y1	y2	y3	y4
x1	0	1	1	0
x2	1	0	0	0
x3	0	1	0	0

Classifier 4

3. **Label Powerset** - We transform the problem into a multi-class problem with one multi-class classifier is trained on all unique label combinations found in the training data.

X	y1	y2	y3	y4
x1	0	1	1	0
x2	1	0	0	0
x3	0	1	0	0
x4	0	1	1	0
x5	1	1	1	1
x6	0	1	0	0

X	y1
x1	1
x2	2
x3	3
x4	1
x5	4
x6	3

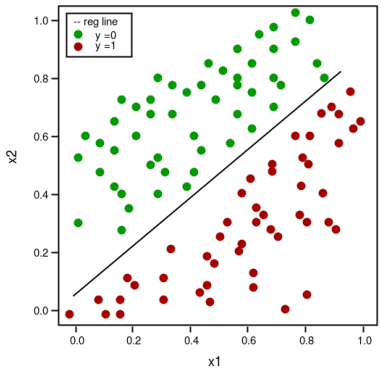
7. MODELS IMPLEMENTED

Supervised Learning
We have used the following supervised learning methods to perform multi-label classification to identify multiple genres for a given movie in our dataset.

7.1 CLASSIFICATION MODELS

1. Logistic Regression Classifier

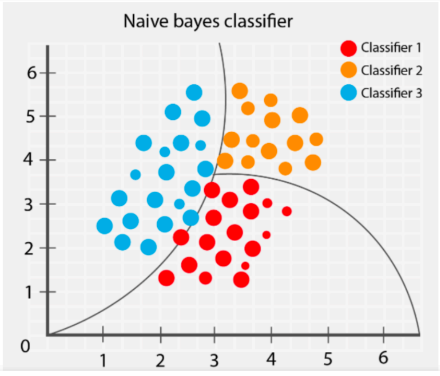
Logistic regression is a classification algorithm that uses supervised learning to predict the probability of a target variable. Because the nature of the target or dependent variable is dichotomous, there are only two possible classes. After splitting the data, we import LogisticRegression from sklearn.linear_model and fit the regressor over the training data.



2. Naive Bayes Classifier

A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem.

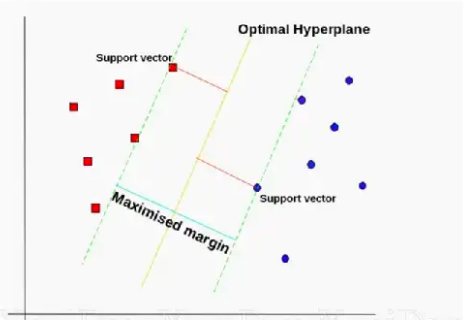
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$



Gaussian Naive Bayes
When the predictors take up a continuous value and are not discrete, we assume that these values are sampled from a gaussian distribution.

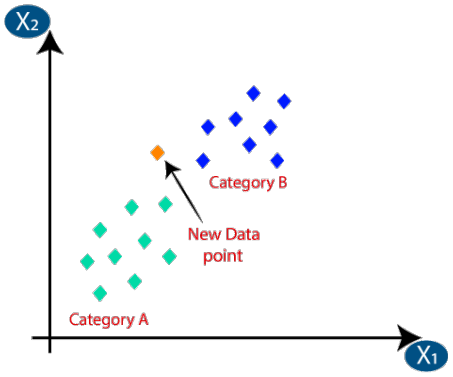
3. Support Vector Machines (SVM)

Support Vector Machines (SVMs) are a type of supervised learning algorithm that can be used for classification or regression tasks. The goal of an SVM is to find the hyperplane in a high-dimensional space that maximally separates the different classes. SVMs are particularly effective in cases where the number of dimensions is greater than the number of samples.



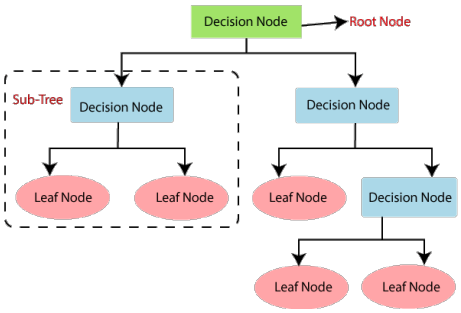
4. KNN classifier

K-Nearest Neighbors (KNN) is a straightforward and effective classification and regression method. It is a lazy, non-parametric learning algorithm. It is non-parametric because it makes no assumptions about the underlying data distribution. A lazy algorithm is one that does not use training data points to generalise. In fact, KNN is known as a lazy algorithm because it does not learn anything from the training data and instead simply stores it.



5. Decision Tree Classifier

A decision tree is a tree structure that looks like a flowchart and is used by an algorithm to make a prediction or decision. It incrementally divides a dataset into smaller and smaller subsets while also developing an associated decision tree. The end result is a tree with leaf nodes and decision nodes.



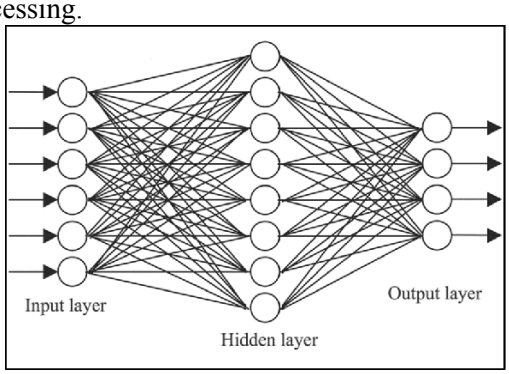
6. AdaBoost Classifier

AdaBoost (Adaptive Boosting) is a simple and effective ensemble method for improving predictive model generalisation. It works by weighting instances in the training dataset based on the base classifier's error, causing the base classifier to focus more on the difficult examples. The ensemble of base classifiers then votes by weighted majority to make the final prediction.

7.2 DEEP LEARNING MODEL

7. Deep Learning Model using Neural Network

Deep learning refers to the use of multiple layers of neural networks, which can learn hierarchical representations of the data. Deep learning models have achieved state-of-the-art results on a wide range of tasks, including image classification, speech recognition, and natural language processing.



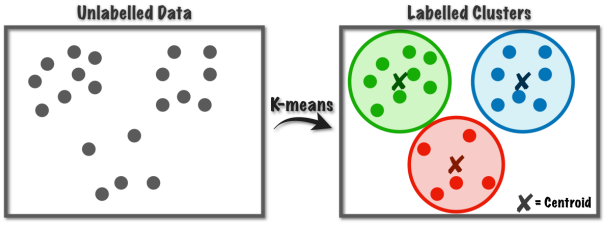
Unsupervised Learning

We have used the following unsupervised learning methods to perform multi-class classification to identify a single genre for a given movie in our dataset at one point.

CLUSTERING MODEL

8. K-Means Clustering

K-Means clustering is an unsupervised learning algorithm that divides a dataset into a specified number of clusters. The goal of K-Means is to partition the data into clusters such that the sum of the distances between the data points and the centroid of the cluster is minimized.



7.3 DIMENSIONALITY REDUCTION MODEL

9. Principal Component Analysis (PCA)

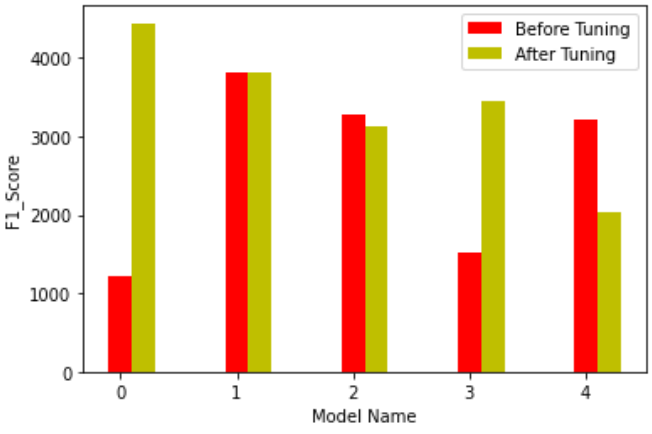
Principal Component Analysis (PCA) is a dimensionality reduction technique that is frequently used to reduce a dataset's complexity while retaining as much variance as possible. This is accomplished by identifying a new set of uncorrelated variables known as principal components that can explain the variance in the data.

7.4 Hyperparameter Tuning

The parameters that define the model architecture are known as hyperparameters, and the process of searching for the best model architecture is known as hyperparameter tuning. Multiple trials are run in a single training job to tune hyperparameters. Each trial is a complete execution of the training application with values for the hyperparameters we specify and set within the limits. Each of the models we implemented have been hyperparameter tuned.

8. MODEL PERFORMANCE

The graph below summarises the comparison between the algorithms both before and after parameter tuning.
The comparison is shown for five different models. The red bar represents the F1 score prior to tuning, while the yellow bar represents the F1 score following tuning.



9. RESULTS

The overall goal of the project is to find one good model for predicting movie genres based on plot summary. We can see that tuning logistic regression results in a higher F1 Score, though tuning is not always good because we simply change the threshold values to achieve a higher F1 Score. The lower the threshold, the less likely the model is to predict correct values while maintaining a high F1 Score. Aside from that, we can also see that KNN provides a good score.

9. CONCLUSION

This project allowed us to gain a better understanding of how machine learning models can be used to understand the semantics of text classification. We discovered that no single set of parameters and layers will work for every problem domain, and that the outcome of a model is highly dependent on the use of an appropriate dataset and proper feature selection. However, the results of our experiments show that our multi-label classification methods are concurrent with the current state-of-the-art approaches.

10. REFERENCES

[1] https://www.analyticsvidhya.com/blog/2017/08/introduction-to-multi-label-classification/

[2] https://github.com/christianversloot/machine-learning-articles/blob/main/creating-a-multilabel-neural-network-classifier-with-tensorflow-and-keras.md

[3] https://www.geeksforgeeks.org/an-introduction-to-multilabel-classification/

[4] https://www.analyticsvidhya.com/blog/2019/04/predicting-movie-genres-nlp-multi-label-classification/

[5] https://medium.datadriveninvestor.com/k-nearest-neighbors-in-python-hyperparameters-tuning-716734bc557f

[6] 'Predicting Movie Genres Based on Plot Summaries'- Quan Hoang, University of Massachusetts-Amherst

[6] https://www.kaggle.com/code/albeffe/text-clustering-tfidf-pca-beginner-tutorial